SPECIFICATION

Please amend the specification as follows:

Please replace the paragraph under the heading, "CROSS-REFERENCED APPLICATION" on Page 1 with the following replacement paragraph:

This application relates to co-pending U.S. patent applications entitled "PSEUDO LRU FOR A LOCKING CACHE" (U.S. Serial No.10/655,366), "IMPLEMENTATION OF A PSEUDO-LRU ALGORITHM IN A PARTITIONED CACHE" (U.S. Serial No. 10/655,401), and "SOFTWARE-CONTROLLED CACHE SET MANAGEMENT WITH SOFTWARE-GENERATED CLASS IDENTIFIERS" (U.S. Serial No. 10/655,365), all filed concurrently herewith. This application also relates to co-pending U.S. patent application entitled "IMPROVED MEMORY MANAGEMENT FOR REAL-TIME APPLICATIONS" (U.S. Serial No. 10/318,541 filed December 12, 2002).

Please replace the paragraph under the heading, "SUMMARY" on Page 2 with the following:

~~The present invention provides a software controlled data replacement for a cache. The software controlled data replacement employs a class identifier and a tag replacement control indicia. The software controlled data replacement for a cache further provides a replacement management table, employable to read the class identifier to create the tag replacement control indicia. The cache comprises a plurality of sets. The cache is employable to disable a replacement of at least one of the plurality of sets as a function of the tag replacement control indicia.~~ The present invention provides a system for managing cache replacement eligibility. A first address register is configured to request an address from an L1 cache. An L1 cache is configured to determine whether a requested address is in the L1 cache and, in response to a determination that a requested address is not in the L1 cache, is further configured to transmit the requested address to a range register coupled to the L1 cache. The range register is configured to generate a class identifier in response to a received requested address and to transmit the requested address and class identifier to a replacement management table coupled to the range register. The replacement management table is configured to generate L2 tag replacement control indicia in response to a received requested address and class identifier. An L2 address register is coupled to the first address register and configured to request an address from an L2 cache. An L2 cache is coupled to the L2 address register and the replacement management table and is configured to determine whether a requested address is in the L2 cache and is further configured to assign replacement eligibility of at least one set of cache lines in the L2 cache in response to received L2 tag replacement control indicia. In response to a determination that a requested address is not in the L2 cache, the L2 cache is further configured to overwrite a cache line within a set of the L2 cache as a function of the replacement eligibility.

Please replace the third paragraph beginning on Page 8 with the following:

In FIGURE 2, the RMT 160 is a software managed table. Software maintains the structure and interprets the meaning of the RMT 160 entries, which specify which cache sets are to be used for specific ClassIDs. In a further embodiment, no hardware checks of the RMT 160 for accuracy are made. Generally, the RMT 160 is employed in the mapping of a missed address range to a set or sets of the L2 cache 170. In one embodiment, the address range associated with a classID is an effective address range. In another embodiment, the address range associated with a classID could be a real or physical address. In the registers 140, 145, the missed address is mapped to a classID. Generally, the classID is associated with one or more memory regions or a given address range. In one embodiment, classID zero corresponds to any address range that is not specifically provided for by the other classIDs. The given classID is then transmitted to and employed as an index to the RMT 160. Using the classID as an index, the RMT 160 is accessed, the replacement information is read, and the tag replacement control indicia are[[is]] generated.

Please replace the second full paragraph beginning on Page 10 with the following:

In the illustrated embodiments, classIDs 5-8 are not used. That is, ~~all entries~~ every entry in each classID is "0". However, those of skill in the art understand that the logical set and classID determinations expressed as "1"s and "0"s, as shown in FIGURE 2, are for purposes of illustration, and that other logical set and classID determinations are within the scope of the present invention, such as through software.

Please replace the first full paragraph beginning on Page 11 with the following:

Accessing an invalid RMT row, according to the "v" valid bit entry, typically results in a default class employed by the RMT 160. In one embodiment, the default class is classID~~ClassID~~ zero. In a further embodiment, accessing an invalid RMT row generates an interrupt to the CPU. In one embodiment, all bits are set for enable for an invalid classID~~ClassID~~ of the RMT 160. In a further embodiment, an error signal is returned to the operating system instead of generating an interrupt. This occurs if the process running on the CPU, but not the operating system itself, is[[are]] terminated.

Please replace the third full paragraph beginning on Page 12 with the following:

Generally, the RSR defines the starting address and the RMR defines the ending address that is to be tested for within the range registers 140, 145. This is performed by the RMR 500 masking the bits of a missed operand's address, and then comparing the missed operand to the bits of the RSR 400. In FIGURE 5A, the upper bits of the RSR 400 pass the starting address of the range to be tested, the lower bits, and the classID. Furthermore, the RMR masking operand does not mask the address range mode (that is, Real or Virtual) bit or[[and]] the valid or disabled bit.

Please replace the second full paragraph beginning on Page 13 with the following:

Turning now to FIGURE 6, illustrated is a classID generator 600. The operand's address is received through input line 610 and an AND 630 performed upon it and the RMR mask 620 for bits 32-51. Then, bits 0-31 are imported from the requested missed address, and bits 32-51 are appended to bits 0-31. These are, in turn, compared to the RSR 640[[400]] in a comparator 650. This comparison is conveyed to an AND 660. The AND 660 receives the equality compare from CMP 650 and the V output of the RSR 640. The V output corresponds to the "valid" bit, and the R output corresponds to the "real" bit.

Please replace the third full paragraph beginning on Page 14 with the following:

In step 750, the classID is read by the RMT 160 and is used as an index for the corresponding row in the RMT 160. For instance, in FIGURE 7[[3]], classID two corresponds to "1,0,1,1,1,0,0,0." In other words, the software defines that for classID two,~~RMT 160~~ sets 0, 2, 3, and 4 are eligible for replacement and sets 5, 6 and 7 are not eligible for replacement.

Please replace the fourth paragraph beginning on Page 14 with the following:

In step 755, tag replacement control indicia is created by the RMT 160. Generally, the tag replacement control indicia is employed to control the replacement eligibility of sets of the L2 cache 170. In step 769[[760]], the requested address is then requested from the L2 cache 170. In step 770, if the L2 cache 170 has the information corresponding to the requested address, then that information is forwarded to the CPU 110 and the method 700 ends in step 775.

Please replace the paragraph under the heading, "ABSTRACT" on Page 21 with the following:

~~The present invention provides for selectively overwriting sets of a cache as a function of a replacement management table and a least recently used function. A class identifier is created as a function of an address miss. A replacement management table is employable to read the class identifier to create a tag replacement control indicia. The cache, comprising a plurality of sets, is employable to disable the replacement of at least one of the plurality of sets as a function of the tag replacement control indicia.~~ The present invention provides a system for managing cache replacement eligibility. A first address register is configured to request an address from an L1 cache. An L1 cache is configured to determine whether a requested address is in the L1 cache and, in response to a determination that a requested address is not in the L1 cache, is further configured to transmit the requested address to a range register coupled to the L1 cache. The range register is configured to generate a class identifier in response to a received requested address and to transmit the requested address and class identifier to a replacement management table coupled to the range register. The replacement management table is configured to generate L2 tag replacement control indicia in response to a received requested address and class identifier. An L2 address register is coupled to the first address register and configured to request an address from an L2 cache. An L2 cache is coupled to the L2 address register and the replacement management table and is configured to determine whether a requested address is in the L2 cache and is further configured to assign replacement eligibility of at least one set of cache lines in the L2 cache in response to received L2 tag replacement control indicia. In response to a determination that a requested address is not in the L2 cache, the L2 cache is further configured to overwrite a cache line within a set of the L2 cache as a function of the replacement eligibility.